

cps-10-forecast

Sun, 5/15 10:25AM 23:30

SUMMARY KEYWORDS

keyboard, firmware, pins, qm, micro, build, python, arduino, chip, keys, circuit, started, controller, feather, columns, design, runs, key, board, support

SPEAKERS

Paul Cutler, Pierre Constantineau

- P** Paul Cutler 00:01
Welcome to The Circuit Python Show. I'm your host, Paul Cutler. This episode, I'm joined by Pierre Constantineau. who owns Blue Micro, which creates and sells custom mechanical keyboards, macro pads and more. Pierre, welcome to the show.
- P** Pierre Constantineau 00:15
Hi, Paul. Thanks for having me.
- P** Paul Cutler 00:17
Glad to have you. Let's start at the beginning. When did you first get into computers or coding?
- P** Pierre Constantineau 00:21
Well, that was actually quite a while ago, my parents, my dad mostly bought me like a ColecoVision Adam. So that was like, early 80s. And then that was a very short lived computer, but it was a computer that I started, like, just basic programming and and then after that, I kind of like abandon it and just got a PC and just started using it and went to like high school, university. And then but I was always the RadioShack catalogs and like, kind of like almost like lusting over like resistors, and so on. But then when I got into college, I actually first got into a course of electronics, and it was probably the first Makerspace and that was like in early 90s. It was kind of like the maker space of a time where you can like learn about just basic electronics and digital chips. And then during that same year, I also did like a Turbo Pascal course, then went on into university. And then at that point, I thought I'd wanted to go like the high school I wanted to be Oh, chemical engineer, and then college. Oh, I want to be electrical engineer then first got into, like, really my engineering degree. And then oh, no, okay, there's this material science course that I think I like even more. I went into material science as a career and wasn't really material science was like metallurgy really. And then it kept on going and doing more

studies and went into more like chemical engineering and when you can look back all of the different studies and interests well now going today, what I do for electronics is really the marriage of a bunch of different things.

P

Paul Cutler 02:01

Let's talk keyboards, you have a tiny shop where you sell a few different kinds of keyboards. How did you get into that?

P

Pierre Constantineau 02:07

Well, it started maybe about four or five years ago, when I first started with like, just I wanted to know about keyboards and mechanical keyboards. And I first started with like just a plain 60% keyboard probably from Aliexpress then moved on into building a split keyboard and at the time, it was like a like split so essentially, a smaller keyboard that's literally split can be a little bit more economic and effect cannot go in it runs on two different controllers. So if you can look inside right here, there pro micros so pro micros really there were there, oh, they're probably still the one of the most common chip for making keyboards, they're getting harder to get because of a shortage these days. But and that's how I got into keyboards. But then a little bit later, there was a like a fellow on Reddit on one of the check mark on one of the mechanical keyboard Reddit that can like say, Oh, here's my split keyboard that uses two feathers and our F 32. Like the original feather. And they made himself a split keyboard, all wireless and like feather like this is actually pretty neat. And I thought, Oh, wow, well, I've got this whole keyboard like that, that let's split that I talked about. And that uses the pro micro. And I thought I gotta make myself something to fit on there. Because like a feather wasn't really used for pretty, like I'd say 99% If you build your own keyboard, it was very likely going to be using a pro micro so I thought, Okay, I gotta build myself something. And that's where I can like first started the blue micro blue for blue suit and micro for for micro size. So that's where the name comes from. And I created literally this, this blue micro here and I just kind of go back into here, literally to make them fit just as a one for one replacement. Obviously, they're bigger because the modules are not very small. And then they're smaller than the like the ESP 32. But they're definitely larger than the pro micro. And then that's when I first really started okay, I'm going to build my own firmware for this because the existing firmware for the AVR boards like the the Arduino based boards, they don't really fit because this is an ARM processor. And even if you look at some of the current firmware that QM K, they might have a branch or an area where all Yeah, we support arm as well but it's very specific chip and if you go outside of that, it's not supported. And then the complicating factor of with these is there's a Bluetooth stack commonage That's a Nordic firmware. It's a bit like a bias for computer Well, there's essentially a bias that is takes care of all the Bluetooth things. So that gets flashed to the chip and the interface to it the saw the software and dev kit, the licensing isn't quite compatible for QM K and lovely library. So there's been more like legal reasons and technical reasons for people saying that I don't want to go in there because it's, it's one of those can of worms that okay, let's not touch it, because it just might be a mess. Or if somebody comes back in, okay, it's not going to be there. So we had to look at okay, what do I do? So I first looked into, okay, let's get like just a plain SDK from Nordic and just try to see if I can build something and it was just impossible is like after like hours of just trying to get a blinky light is like, Ah, okay, that's just too complicated. And, and this kind of looking back on, there's that feather. It's like, it's on

Arduino. Within minutes, I had it's running pretty well. Yeah, minutes, I had it running on here. So so that made it like, Okay, going for hours or work with no success two minutes with, okay, something that flashed on here, I can make a light blink, I can build a whole lot board

P

Paul Cutler 06:20

on this. There's nothing better than that, that first feeling of seeing that blinking light, no matter which programming language you're using, or what board you're using that that sense of accomplishment always feels so good.

P

Pierre Constantineau 07:03

And then after this, it's just really where the blue micro firmware really came. Because I essentially took the I got inspired from QM Kay and it's okay, I do, I want to build something that's flexible enough for other people who want to build their own keyboard to be able to just say, I want to just drop in that blue micro and then just go on and have my own keyboard. So I kind of built on firmware. And then like, at a fair bit of functionality started

P

Paul Cutler 07:31

with our journals, what actually goes into making your own keyboard, look at some of the kits, there's firmware for this. And then there's, there's like the overlay software to help program some of the kids requires soldering, like

P

Pierre Constantineau 07:43

that first code that dad to read it again, and read it had. And it was like, completely different from what it looks like today. But then after that, obviously, the the age 40 came along, which kind of brought in USB, so nowadays, it's like it's like a full keyboard and it's actually pretty neat. So let's just say this is one of the probably my first keyboard that I've actually built with blue micro so it's essentially a 60% as well. And for those that like don't quite understand 60% So you've got an just like, like my wife's keyboard here. So you've got a standard size, which like 110 Key, well remove the numpad and you've got what's called 10 keyless but then remove the function key in the navigation then you get 60% So it's like close enough like 60 keys over 110 Close enough to 60% and then so my first keyboard was literally a 60% that actually plugged in a blue micro underneath and obviously you need a battery because it's wireless. So what this actually this PC board is essentially not even my design it was somebody I don't even know his name but it's like 40% club on the internet so if you just go 40% dot club you'll have a whole blog of somebody who builds just keyboards one off and just documents at all but the actually makes those his design open source and I got myself the keyboard and then soldered in all the diodes and then and then you can see here this is the the format of the pro micro, but how do you go from like a pro micro which that is on here to essentially well 61 keys because if you look carefully pro micro only has eight input output pins means. So going from eight input output pins. This is where the firmware and the hardware, the design of the keyboard really kind of comes in. So the firmware must know, okay, one, which pins are connected where, and then also which in what organization. Let's just take the example of the

the Adafruit macro pad, I think a lot of people will recognize this. Well, this is probably the simplest, if you just disregard the discrete NT. And the rotary encoder, this is probably the simplest because literally what you have is 12 different buttons connected to a controller. So it's a one for one a pin to a key and a key is a button, it's a switch. And it's probably the simplest, because you only have like one to one, it's 12. And then you've got enough pins. How do you go from 12 to 60, or 87, or even 100. While this is where you need to want to have a matrix. And in a matrix, what you do is you essentially have rows and columns. So here's a keyboard with 75 keys, and you've got columns and rows. So each row will be assigned a pin or an input, and then each column will be assigned another pin or another input. Now, typically, the way it works is the matrix, either the rows that are column are going to be an input, the other one's going to be an output. So that this way, you can scan one row at a time, and read the entire set of columns. And then you just say, if I turn on this row, read up columns, and then there's a match up. That's because you have a key here. So what you need to do is to scan over all the rows and all the columns to say, Am I on I'm on my own. And then you have your entire matrix of keys of 100 or 80, or 60. All depending, that's what powerful controllers microcontrollers do is really do all the scanning its contents constantly scanning. And that's the the first part that the firmware must do is to understand what that mapping between how to scan. So is it? Am I slowly scanning just pins or a matrix? And how is that matrix organized? Because obviously, if I have, let's go back to that one here. So if I have 18 pin on the pro, my curl, and then I have 61 keys, okay, well, there's multiple ways you can go and divide those. So 18, it could just say, Okay, well I have 10, could I have eight rows and 10 columns, and then that gives me 80 keys. But then if you count, you've got think 14. So if you do the match, like 14 that's already 14, and you've got four pin left for the rows, but you've got five, that doesn't work. So the way that this actually is organized is it only uses 16. So it's a matrix of eight by eight. And it moves some of the keys around to give you a potential of a 64 keys even leaving you with two extra P to do other things. So if you look this one, I've got a little speaker. So it does beeps. And it's got a MOSFET so it does underglow lights. So it's got all 18 pins used for something.

P

Paul Cutler 13:35

Hi, it's Paul, I'll get you back to the show in just a moment. Thanks for listening. And if you're enjoying the show, please tell a friend to write a review. You can also support the show financially. Your support helps cover the cost of podcast hosting, recording services and transcriptions. For more information visit circuitpython.com/show/support. Now back to the show. So you've been working with Arduino the last couple of years. And then last year, you launched a new keyboard that actually uses Python. What led you to that,

P

Pierre Constantineau 14:02

essentially, last year, like it was a great chip shortage of while and we're still in it. So we can just say 2020 or 2021. But last year, the RP 24 D came came about a lot of the other chips including the NRF 50 twos were getting scars. So I thought okay, I gotta do something a little bit different. And then this RP 24, D link, Raspberry Pi. These made it really, really simple to build your own PCBs and chips because they've got a really great guide on here's how to design your own device with the foot DRP 2040. They've got an essentially a hardware building guide. And they even have a chi CAD set a file that you can use as a reference. So okay, well, I got like, I know how to build keyboards. I know how to design keyboards. I've gotten quite a few.

And it's like okay, well let's just take the RP 24 Where do you just like put it all in? And it all came about and like, for example, like at work, this is like I use my keyboard and it just like it runs. Okay, I don't use the RGB, but it's good RGB and hotswap sockets all the, the nice and 60 things nowadays, but I built into Okay, well, let's make it look available. And it's actually open source, but I do have quite a few available if anybody wants so but it's, it's an I actually created it. And it's like, okay, well with circuit PyCon, because just previously to that, I actually put my second controller on there, like the the blue, micro 840. And it was really simple. So it was like, I really, really can't, can't just how simple Scott and the Adafruit team has really made it simple to take your own controller, take an example that already exists, and then just say, okay, the only differences are these, I push it up. And then I think the most complicated thing was actually getting all the documentation lined up to be able to get my own USB product ID for a disk, because there's open source projects that kind of use those, that was the hardest part. Because getting the keyboard or the controller in in circuit, Python was simple. So the next step would take the keyboard is like, well, I just need to do the same thing. And then that came along, I've got this keyboard running circuit Python. And it was very, very simple to get it on, on there. And then I've even had people Oh, yeah, I want a bigger keyboard. And this is how, like this one came along was somebody came last December and says, Well, are you okay? If I kind of like design like the larger SS? Yeah, sure. Like if you want to contribute it back, and I'd be like, just set up a small production run. So you can have your own and you can like we can test it out, make sure it all works fine. And so this is where the larger 87 came along. And nd boat runs circuit Python, the book, even setting up on QM KMK was very simple, because the examples are really simple. It's just matter, you set up which pins are mapping to which rows in which columns, and then send diskeeper s whenever you detect that this key is attacked. So it's, it was very, very simple.

P

Paul Cutler 17:32

Well, that's great to hear. How is circuit Python helped you with troubleshooting.

P

Pierre Constantineau 17:36

This is where like when I went, going back to Arduino. And if I even go back to the days of QM k, when you look at QM K, let's go with like the hardest, difficult start first, if I want to create my own keyboard, or if I build my own keyboard, just even having my own PCB, you're gonna have a hardware problem. If you can just say, Oh, I have a short between these two pins. And then I've got like press a key and then like there's like three that are showing up at the same time. If you have QM K, that's like all a prepackaged firmware with no troubleshooting tools of any kind, then it's very difficult to figure out what's wrong. Didn't you go to Arduino where, okay, you've got to create a small program, compile it and flash it. And then quite a few people are not comfortable programming. I'd say in the mechanical keyboard community, there's like really two groups. There's those who like the keyboards and that's really what they're in for the keyboard field. And then those there's those leading programmers that yeah, they're not scared of diving in and going programming by selling a number of keyboards and chips like and having that firmware like having troubleshooting tools handy to be able to troubleshoot your own keyboard when something on the hardware does not work well is very useful, actually built a number of tools to be able to help people troubleshoot. Oh, well, Is this thing working? Is it shorted? Is it like just like not connected? So I've had a few tools there, but then I kind of like look at Circuit Python is like, oh my god, like you don't have to compile anything. You just

copied the files. And it's like, is it working? No. Oh, just change a couple of thing. Is it working? Oh, yeah. Okay, finally found it found the problem find solution. It's so much faster to be able to iterate. Because when you troubleshoot, it's always like your mind is working. Okay, do I need a multimeter? Do I need like, try to see what's connected and not connected? Well, if you've got a microcontroller right there for all its spins already probing what you want to like, see, oh, is this connected to this? Well, you can ask these to Ben, are they connected? And then then you'll be able to tell. So it's with circuit Python, it's a whole lot easier to iterate so much faster. err on identifying the root cause of your problems and then moving on into, Okay, I gotta pick up the soldering iron and fix this little piece or that little piece.

P

Paul Cutler 20:11

Well, that's great to hear. We're almost out of time. But before we go, I'm a vinyl record collector and I have a segment called turn the tables, I enjoyed the Pong. I don't know if everyone else does. I've been asking all the questions, now's an opportunity for you to ask me a question. Yeah,

P

Pierre Constantineau 20:25

I'm gonna go a little bit different. Like at a fruit has been going retro lately, with all the floppy disk stuff. But I'm gonna go over a retro question here. If you go back maybe in your childhood, or maybe the first computer you had first computer game. And this you would pay oh, here's the one game that kind of like, okay, I played it from end to end there. It was, like the one thing that really hooked me are kind of like, okay, I've got fond memories of what what would that be?

P

Paul Cutler 20:54

For me, it would be broader bonds Lode Runner on my apple to see. And it was a, you know, kind of a platform scroller, and you could drill a hole and the bad guys would fall in the hole, or you had to drill more holes to go get the gold that was there. But what I loved about it is you could design your own levels. And that was the first game I ever played, that I could design my own levels. And I remember just spending an entire summer just doing that over and over again and making it so making it near impossible to play at times. But it was so much fun. If I had to go back, that would probably be the game that I would choose. Last question, if you're gonna start a new project, which microcontroller Do you think you'll reach for,

P

Pierre Constantineau 21:34

it's gonna be one of two. And I'm gonna have kind of like divided maybe even four ways. So if I just want to prototype something very quickly that I don't want to design, like my own PCB, I would either pick the key B 24. D, because I've got a handful of them. Or if I need more pins, I would probably pick a Pico. But if I need blue toots, I gotta go back to the NRF 50 to 840 it's got, obviously it's got USB as well, but it's got Bluetooth, and I've got quite a few of those of my blue micros, but I could like go for a feather or even go like the bare bone module, which I'm just gonna pull out this one because I actually if I build my own PCB, I could actually go and

say, let's put the modules right on to build my own. Next keyboard. So and this is probably an adventure of kind of like started last November, probably for like a super thin keyboard, like, very, very thin. It was more of an experiment and just sell this. Let's see how low can I get, and it runs again, because it's based on by on that same chip, I was able to load circuit Python and get kmk running on a drill deep quickly. So

P

Paul Cutler 22:56

that's so cool. Well, we're out of time here. Thanks so much for being on the show.

P

Pierre Constantineau 23:00

Thanks again for having me.

P

Paul Cutler 23:02

Thank you to Pierre for being on the show. You can find Pierre on his youtube channel at Blue micro wireless keyboards. You can also find here on GitHub and as Tindie store by searching for blue micro for shownotes transcripts photos and to support the show, visit circuitpython.com. Until next episode, stay positive